React框架

# 一、React开发环境准备

开发环境：node、NPM（安装node后NPM自动安装，下载地址：http://nodejs.cn/download/）

开发工具：VSCODE（注：免费，其他开发工具也可以）

# 二、什么是React

## 1、起源

React 起源于 Facebook 的内部项目，因为该公司对市场上所有 JavaScript MVC 框架，都不满意，就决定自己写一套，用来架设Instagram 的网站。做出来以后，发现这套东西很好用，就在2013年5月开源了。

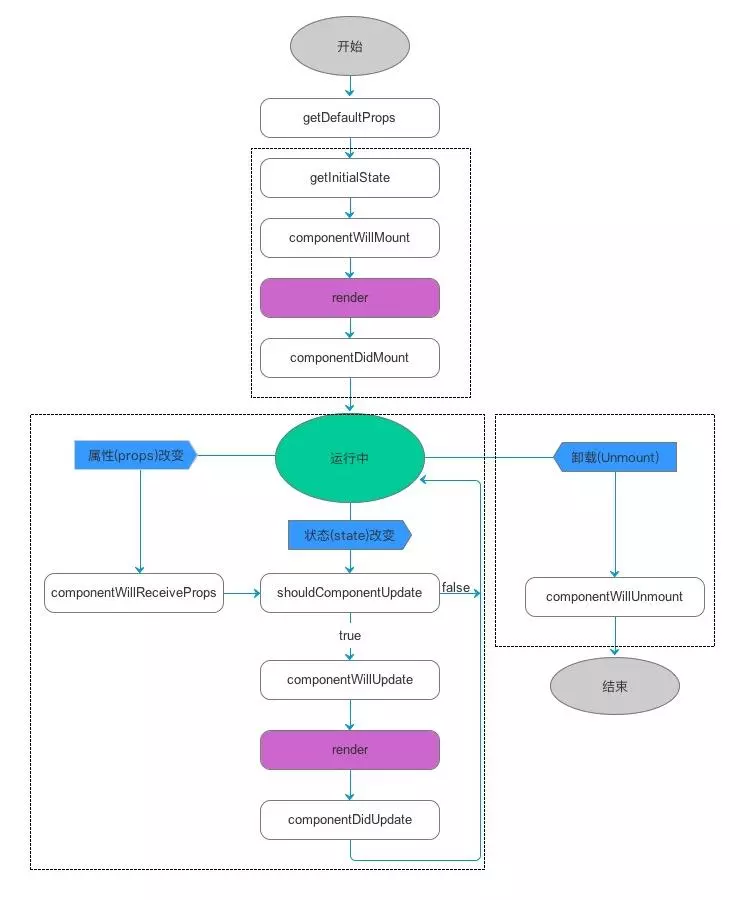
## 2、概念

React是一个用于构建用户界面的轻量级的JavaScript库，也是Web应用程序的视图层, 用于构建高效、快速的用户界面。

React遵循组件设计模式、声明式编程范式和函数式编程概念，以使前端应用程序更高效。它使用虚拟DOM来有效地操作DOM。同时遵循从高阶组件到低阶组件的单向数据流。

# 三、React生命周期

## 1、生命周期



## 2、常用的生命周期

（1）、componentWillMount：页面加载之前，即未渲染DOM前，可以在这里调用ajax请求，返回数据setState后组件会重新渲染；不可以对DOM节点进行操作。

（2）、componentDidMount：页面加载完成，即组件第一次渲染完成，此时DOM节点已经生成，可以在这里调用ajax请求，返回数据setState后组件会重新渲染；可以对DOM节点进行操作。

（3）、componentWillUnmount：离开页面时，通常在此处完成组件的卸载和数据的销毁。

（4）、render：该函数会插入jsx生成的dom结构，react会生成一份虚拟DOM树，在每一次组件更新时，在此react会通过其diff算法比较更新前后的新旧DOM树，比较以后，找到最小的有差异的DOM节点，并重新渲染。因此可以实现更快的局部刷新，提高用户交互效果。

变量：

*// 变量*

  let *// 局部变量*

  var *// 全局变量*

  const *// 常量*

**aHandle** = () => {

    var a = "a";

    let b = "b"

    var list = [1,2,3];

    for (let i = 0; i < list.length; i++) {

      let c = "c";

      a = "a1";

    }

    return a;*// a1*

    reurn b;*// undefined*

  }

赋值：

this.**setState**({

        dataList: canstTreeList || [],

      })

*// dataList===undefined*

    for (let i = 0; i < dataList.length; i++) {

    }

this.state.dataList = [];*// 一定不要这样写，错误的*

获取对象里的内容：

state = {

*// 树*

      dataList: [],*// 常量树*

      dataSource: [],

    }

*// 获取对象里的内容*

    1、this.state.dataList

    2、const { dataList, dataSource  } = this.state;

# 四、使用ajax或fetch

在文件头部引入import $ from 'jquery';

用法与$.ajax一样

const { a, b, c:{ res }} = this.props;

      const { a, b, c} = this.props;

      const { res } = c;

      {

        a: 1,

        b: 2,

        c:{

          res:{}

        }

      }

      c={

        res:{}

      }

Ajax封装：

*// 异步*

return *// 不能用return*

*// 要用回调函数*

*//封装ajax*

global.**ajax** = (params,func) =>{

**console**.**log**("入参：",params);

  let dataObj = params.data;

  if(!global.develop){

    dataObj = {

      ...params.data,

      "appType":'app'

    };

  }

  let traditional = params.traditional||false;

  let async = true;

  if(params.async == false){

    async=false;

  }

  $.**ajax**({

    type: params.type || "POST",*//类型*

    url: global.url+params.url,*//地址*

    xhrFields: {

      withCredentials: true*//携带cookie信息*

    },

    crossDomain: true,

    data:dataObj,*//入参*

    dataType: "json",

    traditional:traditional,*//防止深度序列化*

    async:async,*//*

**success**: function(result, textStatus, request){*//成功回调*

**sessionFunc**(request);

**console**.**log**("出参:",result);

      if(result === null){

        let obj = {code:'1',desc:'未查询到符合条件的数据'};

**func**(false,obj);

        return;

      }

**func**(true,result)

    },

**error**: function(e) {

**console**.**log**("错误回调");

      let obj = {code:'2',desc:'请检查您的网络'};

**func**(false,obj);

    }

  });

}

*//session是否过期，适用于前后端分离*

function **sessionFunc**(request){

  let sessionstatus = request.**getResponseHeader**('sessionstatus');

**console**.**log**("sessionstatus:",sessionstatus);

  if(sessionstatus === 'timeout'){*//会话超时*

    Toast.**info**("会话超时，请重新登录");

    router.**push**('/');

    return;

  }

}

页面调用：

global.**ajax**(

      params,

      function (status,result) {

      },

      function (res) {

      }

     )

*// 箭头函数*

global.**ajax**(params,(status,result)=>{

    });

# 五、React UI 组件库

开发web端推荐使用ant design，链接：<https://ant.design/docs/react/introduce-cn>

开发移动端推荐使用ant design mobile，链接：<https://mobile.ant.design/index-cn>

# 六、父组件与子组件

1、父子组件

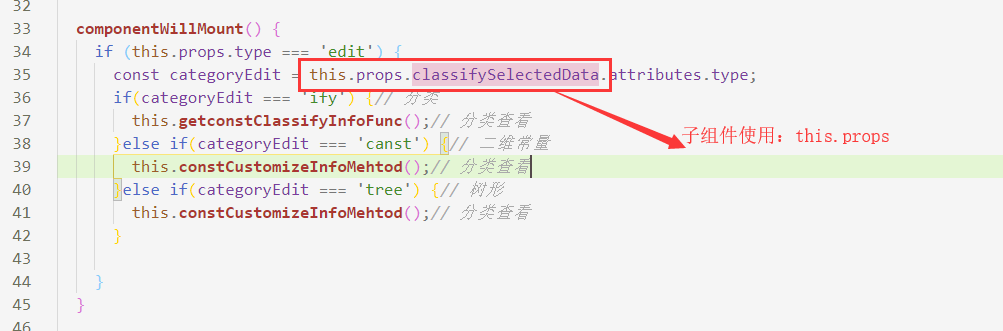


2、父组件调用子组件方法



3、子组件调用父组件方法





# 七、自定义封装组件（进阶）

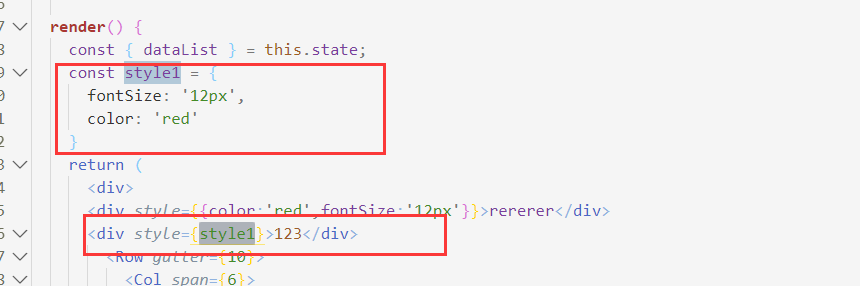
链接：<https://blog.csdn.net/guoyangyang123456/article/details/88100051>

# 八、css

采用驼峰式命名：className



  <div *style*={{color:'red',fontSize:'12px'}}>rererer</div>

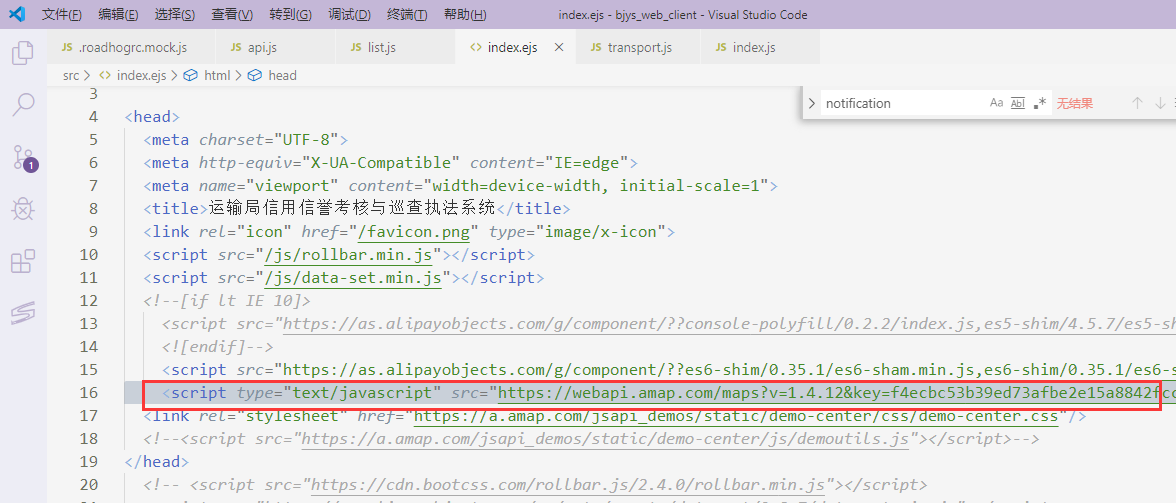


# 九、引入外部js

1、在用到的文件中引入



2、在入口文件引入



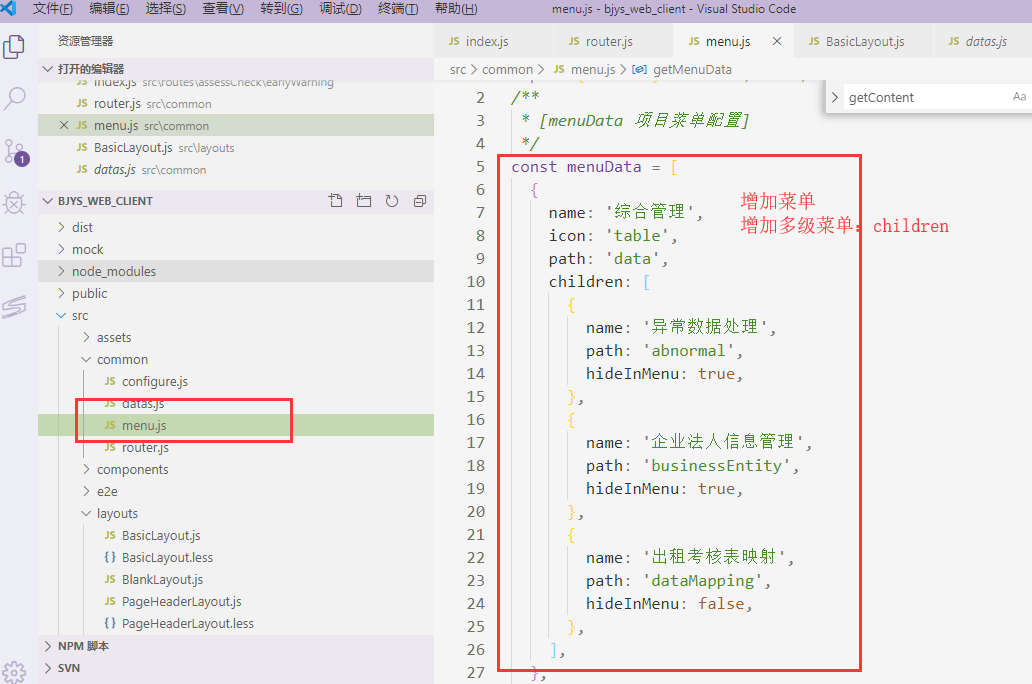
# 十、Jsx语法（了解）

# 十一、路由

 React开发的程序是单页面应用，即一个web项目就只有一个页面（即一个HTML文件）。页面之间的跳转就需要有路由（React Router）的配合。

推荐链接：<http://www.ruanyifeng.com/blog/2016/05/react_router.html>

八、增加新菜单





路由传递参数、页面取参数：<https://blog.csdn.net/luzhaopan/article/details/84996350>

# 十二、常用命令

1、项目基础命令：

npm install 或 cnpm install 或yarn install 下载项目运行所需要的依赖包（开发前）

npm start或npm run dev 运行项目（开发过程中）

npm run dist 或 npm run build 打包项目（将打包生成的文件部署）

2、其他命令：

如果npm太慢，设置淘宝镜像: npm config set registry <https://registry.npm.taobao.org>

使用npm安装cnpm：npm install -g cnpm --registry=https://registry.npm.taobao.org

使用npm安装yarn：npm i yarn -g

安装某个依赖包：npm install jquery --save

# 十三、部署前端程序

将前端程序打包后，部署方式： nginx、 tomcat

nginx参考链接：<https://blog.csdn.net/guoyangyang123456/article/details/83894290>

# 十四、相关知识链接

1. <https://pro.ant.design/index-cn>
2. <http://es6.ruanyifeng.com/>
3. <https://www.runoob.com/react/react-tutorial.html>
4. <https://www.reactjscn.com/docs/hello-world.html>
5. <https://ant.design/index-cn>
6. <http://www.ruanyifeng.com/blog/2016/05/react_router.html>